



# DevOps Shack

## Git Error & Troubleshooting

---

### 1. Merge Conflict

- **Create:**

```
git checkout -b branch1  
echo "Line from branch1" > file.txt  
git commit -am "Change from branch1"
```

```
git checkout main
```

```
git checkout -b branch2
```

```
echo "Line from branch2" > file.txt
```

```
git commit -am "Change from branch2"
```

```
git merge branch1 # This creates a conflict
```

- **Fix:**
  1. Open the conflicting file (file.txt).
  2. Resolve the conflict by keeping the correct lines.
  3. Mark it resolved:

```
git add file.txt
```

```
git merge --continue
```

---

### 2. Push Rejected (Non-Fast-Forward Update)

- **Create:**

1. Clone the repository into two different directories (repo1 and repo2).
2. In repo1, make a commit and push:

```
echo "Change from repo1" > file.txt
```

```
git add file.txt
```

```
git commit -m "Commit from repo1"
```

```
git push
```

3. In repo2, make a conflicting commit and try pushing:

```
echo "Change from repo2" > file.txt
```

```
git add file.txt
```

```
git commit -m "Commit from repo2"
```

```
git push # This will fail
```

- **Fix:**

```
git pull --rebase origin main
```

```
git push
```

---

### 3. Changes Not Staged for Commit

- **Create:**

```
echo "Unstaged change" >> file.txt
```

```
git status
```

- **Fix:**

```
git add file.txt
```

```
git commit -m "Stage and commit changes"
```

---

### 4. File Deleted but Still Tracked

- **Create:**

```
git rm file.txt
```

```
git commit -m "Remove file.txt"
```

- **Fix:**

```
git checkout HEAD file.txt
```

---

### 5. File Ignored but Shouldn't Be

- **Create:** Add file.txt to .gitignore:

```
echo "file.txt" > .gitignore
```

```
echo "Ignored content" > file.txt
```

- **Fix:**

```
git add -f file.txt
```

---

## 6. Accidental Commit to Wrong Branch

- **Create:**

```
git checkout main
```

```
echo "Wrong branch" > file.txt
```

```
git add file.txt
```

```
git commit -m "Commit on the wrong branch"
```

- **Fix:**

```
git checkout correct-branch
```

```
git cherry-pick <commit-hash>
```

```
git checkout main
```

```
git reset --hard HEAD~1 # Undo the commit
```

---

## 7. Forgotten Commit Message

- **Create:**

```
git commit --allow-empty
```

- **Fix:**

```
git commit --amend -m "Add proper message"
```

---

## 8. Corrupted Git Repository

- **Create:**

- Delete or modify .git improperly:

```
rm -rf .git
```

- **Fix:**

```
git init
```

```
git remote add origin <url>
```

```
git fetch origin
```

---

## 9. No Matching Remote Branch

- **Create:**

```
git push origin nonexistent-branch
```

- **Fix:**

```
git push --set-upstream origin <branch-name>
```

---

## 10. Error: Your Local Changes Would Be Overwritten By Checkout

- **Create:**

```
echo "Uncommitted change" >> file.txt
```

```
git checkout main
```

- **Error:** Uncommitted changes prevent switching branches.
- **Fix:**

- **Option 1:** Stash or commit the changes:

```
git stash
```

```
git checkout main
```

- **Option 2:** Discard the changes:

```
git checkout -- file.txt
```

---

## 11. HTTP POST Buffer Error

- **Create:** Attempt to push large files or commit history that exceeds the HTTP post buffer limit.
- **Fix (Temporary, for one session):**

```
git -c http.postBuffer=524288000 push origin main
```

- **Fix (Permanently, global setting):**

```
git config --global http.postBuffer 524288000
```